REMARKS

Reconsideration of the above identified application in view of the preceding amendments and following remarks is respectfully requested.

Claims 1-29 are pending in this application. By this Amendment, Applicant has amended Claims 1, 5, 11, 14, 17, 23 and 25. The claim amendments were made to more precisely define the invention in accordance with 35 U.S.C. § 112, paragraph 2. These amendments have not been necessitated by the need to distinguish the present invention from any prior art. It is respectfully submitted that no new matter has been introduced by these amendments, as support therefor is found throughout the specification and drawings.

In the Office Action, Claims 1-29 were rejected under 35 U.S.C. § 112, second paragraphs, as being indefinite due to minor informalities. Claims 1, 5, 11, 14, 17, 23 and 25 have been amended to correct these informalities. In view of these clarifying amendments, withdrawal of the rejection under 35 U.S.C. § 112 is respectfully requested.

In the Office Action, the Examiner rejected Claim 1-23 and 25-29 under the judicially created doctrine of double patenting. Applicant's representative respectfully requests that the rejection be held in abeyance as a suitable a terminal disclaimer can be filed upon receipt of a Notice of Allowance.

In the Office Action, Claims 1-29 were rejected under 35 U.S.C. § 103 (a) over U.S. Patent No. 5,826,081 to Zolnowsky in view of U.S. Patent No. 5,596,754 to Lomet.

Zolnowsky shows a thread dispatcher for a multiprocessor system (MP). For example, referring to Figure 4, each processor 1, 2, ... N has a separate, respective dispatch queue 401, 402, 403. Each dispatch queue has its own scheduling lock so that any processor attempting to dispatch a thread from a queue needs to acquire a lock for that queue before taking the thread off the queue. Zolnowsky viewed this as an improvement over the prior art where previously a single schedule lock was used for the single dispatch queue for all processors (see col. 6, lines 28-42). Zolnowsky shows nothing related to partitioning data or optimizing locking requirements of data but, rather, merely shows creating processor specific dispatch queues for threads. In short, the dispatch queues of Zolnowsky control the dispatching of threads in a dispatch queue and Zolnowsky is silent with respect to what happens to the target (e.g., locking the entire database) when the thread is executed. In Zolnowsky, the method coordinates and schedules dispatching threads but does not maintain parallel processing because Zolnowsky teaches dispatching threads according to their priority (see col. 8, lines 65-67).

Lomet describes a private lock management method in which multiple servers access common data without frequent calls upon the global lock manager to mediate data requests. Two primary distributed database systems are disclosed, a "shared nothing" system and a "data sharing" system. In a shared nothing system, each subset (partition) of the data is accessed by only one server at a time. All updates are done by a single server specifically assigned to a particular partition. This single server also performs lock management such that other servers cannot update the data within the partition. In effect, each server does local locking for its respective partition. In a data sharing system, multiple servers can access all data at the same time. Although Lomet mentions partitions, there is no teaching, motivation or suggestion as to how such partitions are formed.

There is nothing in either of these references that discloses or suggests, either alone or in combination, in whole or in part, the device defined by Claims 1, 11 and 25 of the subject application. In particular, there is nothing in either Zolnowsky or Lomet which discloses or suggests, a method for reducing contention of a highly contended software lock protecting data items of a data set, including, *inter alia*, the steps of partitioning at least a portion of the data items that are accessible by the identified code path into the partitions and optimizing locking requirements of the identified code path so the locks being acquired and released in the identified code path are those associated with the data items being accessible by the identified code path. To the contrary, Zolnowsky and Lomet teach nothing about how to partition the data set or optimize locking requirements associated therewith. Accordingly, Claims 1, 11, 25 and each of the claims depending therefrom are not rendered obvious by the combination of references cited by the Examiner, and withdrawal of the rejection under 35 U.S.C. §103 (a) is respectfully requested.

Turning to Claim 17, there is nothing in either of these references that discloses or suggests, either alone or in combination, in whole or in part, the device defined by Claim 17. In particular, there is nothing in either Zolnowsky or Lomet which discloses or suggests, a method for reducing contention of a highly contended software lock protecting data items of a data set, all of the data items being stored in a system memory of a multi-processor computer system. The method includes the steps of defining partitions within the data set, creating N partition locks, one for each partition, where N > 2, partitioning at least one of the data items into the partitions and modifying locking requirements of all code paths of the one or more code paths of a software program that access one or more of the data items so that the locks being acquired and released in each of said all code paths are those associated with the accessible data items of each respective partition. As a result, the data items that are accessible by the code paths determine the modifications to the locking requirements. The combination of references have no such teaching. Thus, Claim 17 and each of the claims depending therefrom are not rendered obvious by the combination of references cited by the Examiner, and withdrawal of the rejection under 35 U.S.C. §103 (a) is respectfully requested.

Turning to Claim 23, there is nothing in either of these references that discloses or suggests, either alone or in combination, in whole or in part, the device defined by Claim 23. In particular, there is nothing in either Zolnowsky or Lomet which discloses or suggests, a method for reducing contention of a highly contended software lock protecting data items of a data set. The method includes, *inter alia*, the steps of partitioning a plurality of the data items into the partitions, modifying locking requirements of all code paths of the one or more code paths of a software program that access one or more of the data items so that the locks being acquired and released in each of the all code paths are those associated with accessible data items and evaluating the software program after modifying the locking requirements so as to determine if overall performance of the software program is acceptable, wherein the overall system performance is based on reducing contention of the highly contended software lock. Consequently, Claim 23 and each of the claims depending therefrom are not rendered obvious by the combination of references cited by the Examiner, and withdrawal of the rejection under 35 U.S.C. §103 (a) is respectfully requested.

Any additional fees or overpayments due as a result of filing the present paper may be applied to Deposit Account No. 04-1105. It is respectfully submitted that all of the claims now remaining in this application, namely Claims 1-29, are in condition for allowance, and such action is earnestly solicited.

If after reviewing this amendment, the Examiner believes that a telephone interview would facilitate the resolution of any remaining matters the undersigned attorney may be contacted at the number set forth herein below.

Respectfully submitted,

Date: October 6, 2004

George Chaclas

George N. Chaclas, Reg. No. 46,608
Edwards & Angell LLP
Attorney for Applicants
P.O. Box 55874
Boston, MA 02205
Tel: (401) 276-6653
Fax: (888) 325-1684
Email: gchaclas@edwardsangell.com